The FMS MOM4 User Guide

Lori Thompson <Lori.Thompson@noaa.gov>

Giang Nong <Giang.Nong@noaa.gov>

Table of Contents

FAQs [http://www.gfdl.noaa.gov/~gtn/mom4_faq.html]
MOM4beta3 email (frozen) [mom4beta3_email.html]

1. Introduction

1.1. What is MOM?

The Modular Ocean Model (MOM) is a numerical representation of the ocean's hydrostatic primitive equations, and it is designed primarily as a tool for studying the global ocean climate system. MOM4 is the latest version of the GFDL ocean model whose origins date back to the pioneering work of Kirk Bryan and Mike Cox in the 1960s-1980s. It is developed and supported by researchers at NOAA's Geophysical Fluid Dynamics Laboratory (GFDL [http://www.gfdl.noaa.gov]), with contributions also provided by researchers worldwide.

The first release of MOM4 took place January of 2004. There have been four releases of this code: MOM4p0a (Jan 2004), MOM4p0b (March 2004), MOM4p0c (August 2004), and MOM4p0d (May 2005). The developers welcome feedback, both positive and negative, on the code's integrity and portability as well as documentation. It is through such feedback that the code and documentation evolves and becomes more robust and user friendly.

The purpose of this web guide is to provide general information about MOM4 and particular information for how to download and run the code.

1.2. MOM4 registration

MOM4 users can acquire the source code and associated datasets from GForge [http://fms.gfdl.noaa.gov/], and are required to register [https://fms.gfdl.noaa.gov/account/register.php] at the GFDL GForge location [http://fms.gfdl.noaa.gov/]. Therefore, users need to register only once to get both the source code and datasets of MOM4. More details can be found in the quickstart_guide.html.

1.3. MOM4 email list

Email concerning MOM4 should be directed to the mom4-email list located at `<oar.gfdl.mom4p0@noaa.gov>`. All questions, comments, and suggestions are to be referred to this list. An archive of all emails is maintained at the mom4 email archive [http://fms.gfdl.noaa.gov/mail/?group_id=5]. A subject-organized archive of the emails is available at mom4p0_email.html. Note that by registering at GForge to access the code, you are automatically subscribed to the email list.

1.4. MOM4p0: May 2005

There have been four releases of MOM4p0: (1) MOM4p0a was released January 2004, (2) MOM4p0b was released March 2004, and (3) MOM4p0c was released August 2004, and (4) MOM4p0d released May 2005. Each release is documented on this web page.

1.5. Ongoing issues

There are two major algorithm issues that are being addressed by GFDL developers. (1) Generalized quasi-Eulerian vertical coordinates are being implemented in MOM4. It is anticipated that MOM4p1a will have this capability, with a release date hopefully late 2005. (2) Two-way nesting is being considered for MOM. It is hoped that nesting will be available also late 2005 or early 2006.

There remains two general ways to compile mom4: with static allocation of arrays or dynamic allocation. Recent work on the SGI machines at GFDL has reduced the difference in efficiency between these two compilations. We understand that on some platforms, the dynamic allocation actually is better than static. Such remains an ongoing issue, as do other elements of code efficiency. Our general goal is to provide code that is efficient across a broad range of computer platforms, but not at the cost of sacrificing portability. Those who know they will be working with one particular platform for a period of time should readily find better ways of coding some parts of mom4 and the associated FMS code. If you feel your efficiency improvements are of a general nature and wish to have them distributed in future MOM4 releases, we would be happy for you to contribute the modified code.

1.6. The mom4 community

Since its release in January 2004, there have been nearly 300 registrations with the mom4 distribution, with each registration generally representing more than one user. This is a sizable user community. For various purposes, it is useful to tabulate these users. Click here [http://www.gfdl.noaa.gov/~gtn/mom4_users.html] to download an updated table of users.

2. Details of MOM4

2.1. Documentation

In addition to this user guide, documentation for MOM4 is provided by two LaTeX generated postscript documents:

1. A Technical Guide to MOM4 [http://www.gfdl.noaa.gov/~smg/pointers/geo_physics_abstracts/guide.pdf] by `<Stephen.Griffies@noaa.gov>`, `<Matthew.Harrison@noaa.gov>`, `<Ronald.Pacanowski@noaa.gov>`, and `<Tony.Rosati@noaa.gov>`. This is the primary reference for MOM4. It contains details about some of the numerical algorithms and diagnostics. All usage of MOM4 in the literature should refer to this document:

    > A Technical Guide to MOM4
    > GFDL Ocean Group Technical Report No. 5
    > S.M. Griffies, M.J. Harrison, R.C. Pacanowski, and A. Rosati
    > NOAA/Geophysical Fluid Dynamics Laboratory
    > Available on-line at http://www.gfdl.noaa.gov/~fms.

2. A theoretical rationalization of ocean climate models is provided by Fundamentals of Ocean Climate Models [http://pup.princeton.edu/titles/7797.html]. This book has been published by Princeton University Press, August 2004.

2.2. Complete tree of embedded documentation

The documentation of most Fortran modules in FMS is inserted right in the source code to ensure consistency between the code and documentaion. A Perl software is used to extract documentation from the source code to create a corresponding .html module. For example, documentation for shared/diag_manager/diag_manager.F90 module is shared/diag_manager/diag_manager.html. In general, the embedded documentation is a good starting point to understand the Fortran module. Here is a complete tree [http:/ / www.gfdl.noaa.gov/ ~gtn/ module_doc.html] that depicts all documentation of MOM4 code as well as shared code.

2.3. Characteristics

Although MOM4 shares much in common with earlier versions of MOM, it possesses a number of computational, numerical, and physical characteristics that are noteworthy. The following provides an overview of the main characteristics of MOM4 (please refer to A Technical Guide to MOM4 [http://www.gfdl.noaa.gov/~smg/pointers/geo_physics_abstracts/guide.pdf] for references).

Computational characteristics of MOM4 include the following.

- MOM4 is coded in Fortran 90 and physical units are MKS.
- MOM4 meets the code standards set by the GFDL Flexible Modeling System (FMS [http:/ / www.gfdl.noaa.gov/~fms]). It also utilizes a substantial number of software infrastructure modules shared by other FMS-based models. In particular, all I/O (e.g., restarts, forcing fields, initial fields) is handled via NetCDF [http://www.unidata.ucar.edu/packages/netcdf/].
- There is only a single cpp-preprocessor option (i.e., `ifdefs`) associated with the handling of memory in the model. All options for physical and dynamical choices are handled via namelists and/or switching in/out of modules at compile time. Removing `ifdefs` allows for more readable code that possesses a higher level of error checker handling.
- 2D (latitudinal/longitudinal) horizontal domain decomposition is used for single or multiple parallel processors. Correspondingly, 3D arrays are dimensioned (i,j,k) instead of (i,k,j) used in earlier MOMs. MOM4 has no memory window or slabs.

Numerical and kinematic/dynamic characteristics of MOM4 include the following.

- Generalized orthogonal horizontal coordinates are used. GFDL is supporting both the standard spherical coordinates as well as the "tripolar" grid of Murray (1996). Details are provided in A Technical Guide to MOM4 [http://www.gfdl.noaa.gov/~smg/pointers/geo_physics_abstracts/guide.pdf].
- Bottom topography is represented using the partial cells of Pacanowski and Gnanadesikan (1998). The older full cell approach is available as a namelist in the preprocessing code used to generate the grid specification file.
- The dynamics/kinematics for MOM4p0a and MOM4p0b are based on the non-Boussinesq method of Greatbatch et al. (2002). The Boussinesq option is available via a namelist. This option was jettisoned in MOM4p0c for purposes of code brevity, and in anticipation of using pressure coordinates in MOM4p1 to realize non-Boussinesq dynamics.
- The time tendency for tracer and baroclinic velocity can be discretized two ways. (1) The first approach uses the traditional leap-frog method along with a Robert-Asselin time filter. Note that the Euler forward or Euler backward mixing time step used in earlier MOMs has been eliminated. This method is avalable in MOM4p0a, MOM4p0b and MOM4p0c. (2) MOM4p0c provides an additional method, which is strongly recommended. Here, the time tendency is discretized with a two-level forward step, which eliminates the need to time filter. Tracer and velocity are staggered in time, thus providing second order accuracy in time. For certain model configurations, this scheme is roughly twice as stable as the leap-frog, thus allowing for a factor of two in computational savings. Without the time filtering needed with the leap-frog, the new scheme conserves total tracer to within numerical roundoff. This scheme shares much in common with time stepping used in the Hallberg Isopycnal Model and the MIT GCM. It is the default in MOM4p0c. Details of both the leap-frog and two-level schemes are provided in Fundamentals of Ocean Climate Models [http:/ / pup.princeton.edu/titles/7797.html].
- The sole external mode solver is a variant of the Griffies et al. (2001) explicit free surface. There are two time stepping schemes supported: (1) leap-frog and (2) predictor-corrector. The predictor-corrector is more stable and is thus the default method. Top model grid cells have time dependent volume, thus allowing for total

tracer to be conserved to within roundoff. The linearized free surface method used in MOM3 (and many other implementations of the free surface in z-models) has been jettisoned since it precludes tracer conservation. Details are provided in Fundamentals of Ocean Climate Models [http://pup.princeton.edu/titles/7797.html].

- McDougall et al. (2003) equation of state has been implemented, with in situ density a function of the local potential temperature, salinity, and hydrostatic pressure (baroclinic pressure plus free surface pressure plus applied pressure from the atmosphere and sea ice). Details are provided in Fundamentals of Ocean Climate Models [http://pup.princeton.edu/titles/7797.html].
- Tracer advection is available using various schemes. The centered 2nd, 4th, 6th order schemes are available, as documented in The MOM3 Manual of Pacanowski and Griffies (1999) [http://www.gfdl.gov/~smg/MOM/web/guide_parent/guide_parent.html]. The 4th and 6th order schemes assume constant grid spacing, which simplifies the code though compromises accuracy on the more commonly used non-uniform grids. The Quicker scheme documented by Holland et al. (1998) and The MOM3 Manual [http://www.gfdl.gov/~smg/MOM/web/guide_parent/guide_parent.html] is available. Finally, two multi-dimensional flux limited schemes have been ported from the MIT GCM. These schemes are monotonic and have been found to be roughly the same cost as the Quicker scheme. GFDL researchers have found the Sweby scheme to be most satisfying for many applications, such as biogeochemistry. Hence, effort has been made to enhance this scheme's efficiency in MOM4p0c. Other schemes remain much as they were with the MOM4p0a release.
- Tidal forcing from the various lunar and solar components are available to force the free ocean surface.
- Open boundary conditions are available to allow open boundaries in either of the north, south, east, or west directions.

Physical parameterizations available in MOM4 include the following.

- Neutral tracer physics includes Redi neutral diffusion according to Griffies et al. (1998), and Gent-McWilliams stirring according to the Griffies (1998) skew-flux method. Two-dimensional flow dependent diffusivities are available and can be determined in many different ways, such as the depth integrated Eady growth rate and Rossby radius of deformation, as motivated by the ideas of Held and Larichev (1996) and Visbeck et al. (1997). Details are provided in Fundamentals of Ocean Climate Models [http://pup.princeton.edu/titles/7797.html].
- Vertical mixing schemes include the time-independent depth profile of Bryan and Lewis (1979), the Richardson number dependent scheme of Pacanowski and Philander (1981), and the KPP scheme of Large et al. (1994).
- Horizontal friction schemes include uniform and grid dependent viscosity schemes, as well as the shear-dependent Smagorinsky viscosity according to Griffies and Hallberg (2000). The anisotropic scheme of Large et al. (2001) and Smith and McWilliams (2002) has been implemented. Details are provided in Fundamentals of Ocean Climate Models [http://pup.princeton.edu/titles/7797.html].
- Topographically oriented tracer diffusion introduces enhanced diffusion when heavy parcels are above lighter parcels. It is implemented according to the ideas of Beckmann and Döscher (1997) and Döscher and Beckmann(1999). Details are provided in A Technical Guide to MOM4 [http://www.gfdl.noaa.gov/~smg/pointers/geo_physics_abstracts/guide.pdf].
- The "overflow" scheme of Campin and Goosse (1999) has been implemented, whereby gravitationally unstable fluid parcels are allowed to move downslope via an upwind advection scheme.
- Penetration of shortwave radiative heating into the upper ocean is generally attenuated by the inclusion of chlorophyll data.

Miscellaneous features of the code released with MOM4 include the following.

- MOM4 comes with three tracer packages: (1) ideal age tracer (2) tracers for the OCMIP biotic protocol (3) CFC tracers. Additionally, a suite of code is available for handling tracers inside MOM4 and FMS shared code (`field_manager` [`../src/shared/field_manager/field_manager.html`]). This code provides the user with many options for adding new tracer packages, ecosystem models, etc.

- MOM4 has numerous diagnostics for checking algorithm and solution integrity. These diagnostics include budgets for energetic consistency, tracer conservation, solution stability, etc. Additional diagnostics are available for numerous fields of relevance to the different physics schemes, as well as term balances.
- MOM4 is distributed with a prognostic sea ice model (SIS) via mom4_test4 and mom4_test5.
- As a forward model, MOM4 is compatible with the most recent automatic differentiation tool of Ralf Giering and Thomas Kaminski. See their FastOpt home page at http://www.FastOpt.com/frameset.html for more details.

2.4. MOM4 and FMS

MOM4 has been coded within GFDL's Flexible Modeling System (FMS [http://www.gfdl.noaa.gov/~fms]). Doing so allows for MOM4 developers to use numerous FMS infrastructure and superstructure modules that are shared amongst various atmospheric, ocean, sea ice, land, vegetative, etc. models. Common standards and shared software tools facilitate the development of high-end earth system models, which necessarily involves a wide variety of researchers working on different computational platforms. Such standards also foster efficient input from computational scientists and engineers as they can more readily focus on common computational issues.

The following list represents a sample of the FMS `shared` modules used by MOM4.

- `time manager` [../src/shared/time_manager/time_manager.html]: keeps model time and sets time dependent flags
- `coupler` [../src/coupler/coupler_main.html]: used to couple MOM4 to other component models
- `I/O` [../src/shared/mpp/mpp_io.html]: to read and write data in either NetCDF, ASCII, or native formats
- `parallelization tools` [../src/shared/mpp/mpp.html]: for passing messages across parallel processors
- `diagnostic manager` [../src/shared/diag_manager/diag_manager.html]: to register and send fields to be written to a file for later analysis
- `field manager` [../src/shared/field_manager/field_manager.html]: for integrating multiple tracers and organizing their names, boundary conditions, and advection schemes

The FMS infrastructure (the "Lima version") has been released to the public on GForge [http://fms.gfdl.noaa.gov/], with further releases every few months.

2.5. Test cases

MOM4 is distributed with a set of test cases located in `mom4/exp/`. These tests are taken from models used at GFDL [http://www.gfdl.noaa.gov] for testing the numerical and computational integrity of the code.

Warning

These experiments are *NOT* sanctioned for their physical relevance. They are instead provided for the user to learn how to run MOM4, and to verify the numerical and/or computational integrity of the code. *PLEASE* do not assume that the experiments will run for more than the short time selected in the sample runscripts.

`mom4_test1`: This experiment consists of a flat bottom sector domain and uses very simple physics. This experiment is very small and can be easily run on a single workstation. It should provide the user with a basic experience of running mom4.

`mom4_test2`: This experiment consists of a flat bottom sector domain and it uses more realistic physics options. This experiment has the same grid as `test1`, but it more thoroughly exercises the model's various physics packages. Since the domain is small, it should be accessible to those with single workstations, though it will be slower than mom4_test1 due to the use of more realistic physical parameterizations.

`mom4_test3A`: This experiment consists of an east-west channel using open boundary conditions at the western end of the channel. This experiment provides an illustration of the open boundary condition capability of MOM4. Tracers include potential temperature and salinity.

`mom4_test3B`: This experiment consists of an solid wall on all four sides and with a domain twice as long as the open boundary condition `test3A`. This experiment is used to verify the relevance of the open boundary solution from `test3A` solution at early times. That is, for early times, the solution from `test3A` should closely (though not bitwise) agree with that from `test3B`. Tracers include potential temperature and salinity.

`mom4_test4`: This model uses a global tripolar grid with roughly "3-degree" resolution and 28 vertical levels. The ocean is coupled to the GFDL sea ice model. Tracers include temperature and salinity, as well as the age, OCMIP biotic, and cfc tracer packages. This model is aimed to test the capabilities of the multipe tracers available with mom4. Note that the Sweby advection scheme is used here for all tracers, as is recommended by GFDL researchers.

`mom4_test5`: This model uses a global tripolar grid with roughly "1-degree" resolution and 50 vertical levels. The ocean is coupled to the GFDL sea ice model. The configuration is forced with the German OMIP dataset. Ocean tracers include potential temperature and salinity as well as the age tracer package. This is a large model, and it is similar (though not the same) to the ocean and ice configuration used for the GFDL IPCC simulations.

`mom4_iom`: This is Indian Ocean Community Model version 1.0 developed for Intensive Course on Ocean Modeling held at CMMACS, India, October 4-14, 2004.

3. Contributing MOM4-modules

As with previous MOMs, the GFDL-MOM4 developers aim to provide the international climate research community with a repository for robust and well documented methods to simulate the ocean climate system. Consequently, we encourage researchers to support various modules that are presently absent from MOM4, yet may arguably enhance the simulation integrity (e.g., a new physical parameterization or new advection scheme) or increase the model's functionality.

Depending on the level of code contributions, we envision a directory where "contributed MOM4 code" will reside. Maintenance and ownership of this code will remain with the contributor. As a practical matter, prior to spending time developing a new module, it is recommended that the developer query the MOM4 mailing list [mailto:oar.gfdl.mom4p0@noaa.gov] to see what efforts in the community may have already been availed.

Requirements that contributed code must meet include the following:

1. Clean modular Fortran 90 code that minimally touches other parts of the model
2. Satisfaction of the FMS [http://www.gfdl.noaa.gov/~fms] code specifications outlined in the FMS Developers' Manual [http://www.gfdl.noaa.gov/~vb/FMSManual/FMSManual.html]
3. Compatibility with the MOM4 test cases [#mom4 test cases]
4. Thorough and pedagogical documentation of the module for inclusion in A Technical Guide to MOM4 [http://www.gfdl.noaa.gov/~smg/pointers/geo_physics_abstracts/guide.pdf] (a Latex document)
5. Comments within the code emulating other parts of the model so that HTML documentation files can be generated by our converter

4. Source code and data sets

4.1. Obtaining source code and data sets

The FMS [http://www.gfdl.noaa.gov/~fms] development team uses a local implementation of GForge [http://fms.gfdl.noaa.gov] to serve FMS software, located at http://fms.gfdl.noaa.gov. In order to obtain the source code and data sets, you must register [https://fms.gfdl.noaa.gov/account/register.php] as an FMS user on our software server. After submitting the registration form on the software server, you should receive an automatically generated confirmation email within a few minutes. Clicking on the link in the email confirms the creation of your account.

After your account has been created, you should log in [https://fms.gfdl.noaa.gov/account/login.php] and request access to the Flexible Modeling System [http://www.gfdl.noaa.gov] project. Once the FMS project administrator grants you access, you will receive a second email notification. This email requires action on the part of the project administrator and thus may take longer to arrive. The email will contain a software access password along with instructions for obtaining the release package, which are described below.

To check out the release package containing source code, scripts, and documentation via CVS, type the following commands into a shell window. You might wish to first create a directory called `fms` in which to run these commands. You should enter the software access password when prompted by the **cvs login** command. At cvs login, the file `~/.cvspass` is read. If this file does not already exist, an error message may display and the cvs login may fail. In this event, you should first create this file via **touch ~/.cvspass**.

```
cvs -z3 -d:pserver:cvs@fms.gfdl.noaa.gov:/cvsroot/mom4 login
   (Enter the password you receive in the welcome email here)
cvs -z3 -d:pserver:cvs@fms.gfdl.noaa.gov:/cvsroot/mom4 co -r mom4p0d_patch20050717
```

This will create a directory called `mom4` in your current working directory containing the release package.

If you have already checked out this CVS tag previously and some files have been modified since your last checkout you may wish to update your code as follows:

```
cvs -z3 -d:pserver:cvs@fms.gfdl.noaa.gov:/cvsroot/mom4 login
   (Enter the password you receive in the welcome email here)
cvs -z3 -d:pserver:cvs@fms.gfdl.noaa.gov:/cvsroot/mom4 update -r mom4p0d mom4
```

If you prefer not to use CVS, you may download the tar file from https://fms.gfdl.noaa.gov/projects/mom4/.

Sample output is also available there for download. See Section 8.1, “Sample model output” for more information on the sample output.

All data sets that are needed to run MOM4 test cases [#mom4 test cases] are available for download from the same place in GForge where users get the source code. Therefore, users need to register only once to get both the source code and data sets of MOM4. More details can be found in the quickstart_guide.html.

4.2. Description of the data sets

The topography data set for `test4`, and `test5` is a coarsened version of that kindly provided by Andrew Coward and David Webb at the Southampton Oceanography Centre [http://www.soc.soton.ac.uk/JRD/OCCAM/welcome.html]. Their topography is a montage of that developed by Smith and Sandwell [http://topex.ucsd.edu/marine_topo/mar_topo.html] (1997) by satellite data in the region of 72°S to 72°N, the NOAA (1988) 5-minute global topography ETOPO5 [http://www.ngdc.noaa.gov/mgg/global/etopo5.HTML], and the International Bathymetric Chart of the Arctic Ocean (IBCAO [http://www.ngdc.noaa.gov/mgg/bathymetry/arctic/arctic.html]). The chlorophyll-a density data set was compiled by Colm Sweeney [mailto:Colm.Sweeney@noaa.gov], using data from James A. Yoder [http://www.gso.uri.edu/faculty/yoder.html] and Maureen A. Kennelly at the Graduate School of Oceanography [http://www.gso.uri.edu/], University of Rhode Island [http://www.uri.edu/]. This data set contains monthly chlorophyll concentrations from the SeaWiFS [http://seawifs.gsfc.nasa.gov/SEAWIFS.html] satellite for the period 1999-2001. Monthly wind stress is based on Hellerman and Rosenstein (1983). Temperature and salinity initial and boundary conditions are provided by the NOAA [http://www.noaa.gov] National Oceanographic Data Center (NODC [http://www.nodc.noaa.gov/]) World Ocean Atlas (WOA [http://www.nodc.noaa.gov/OC5/indprod.html]).

All datasets of MOM4 are in NetCDF [http://www.unidata.ucar.edu/packages/netcdf/] format since this format is widely used in the community. A number of useful tools are available here [http://nco.sourceforge.net/] that allow the user to perform some necessary operations (editting attributes, merging, etc.) on a NetCDF file.

5. Setting up an experiment with mom4

MOM4 is distributed with code used to generate model grids, initial conditions, and boundary conditions. Each step must be performed prior to running the ocean model. The steps used during this experimental setup stage are generally termed "preprocessing", and the code used for these purposes is under the /preprocessing directory in the mom4 distribution. The purpose of this section of the User Guide is to outline this code and its usage. Further details of usage and algorithms can be found in the internal documentation within the various preprocessing code modules.

5.1. General comments

We start this section with some general comments regarding the setup of a model experiment.

--Setting up an experiment is critical part to the success of a research or development project with mom4. It is important that the user take some time to understand each of the many steps, and scrutinize the output from this code.

We have endeavoured over the years to provide tools facilitating the ready setup of a new experiment. However, we remain unable to provide code that does everything possible under the sun. Additionally, all features that are provided here may not be fully tested. For these reasons, the preprocessing code continues to evolve as use and functionality evolve. We sincerely appreciate ALL comments about code and documentation, especially comments regarding clarity, completeness, and correctness. Your input is essential for the improvement of the code and documentation.

--Many steps in idealized experiments that were formerly performed while running earlier MOM versions have been extracted from mom4 and placed into preprocessing. Hence, even if you are running an idealized experiment, it is likely that you will need to perform some if not all of the preprocessing steps discussed here.

--In addition to this section discussing how to set up an experiment, the online USER GUIDE has a Frequently Asked Questions (FAQ) [http://www.gfdl.noaa.gov/~gtn/mom4_faq.html] section devoted to these issues. If you have a problem that is not addressed either here or the FAQ, then please feel free to query the mom4 email list. No question is too silly, so please ask!

--All code used to setup an experiment with mom4 is written in Fortran 90/95 except make_xgrids, which is written in C. Most code is dependent on FMS shared code for the purpose of parallization and interpolation. In addition to the documentation provided here and FAQs [http://www.gfdl.noaa.gov/~gtn/mom4_faq.html], there are comments within the code to help users debug and to make modifications to suit their purpose.

--Some users make nontrivial changes of general use. With your support, assistance, and maintenance, we will endeavour to include your changes in future releases.

5.2. Creation of the ocean/ice grid

Within GFDL FMS, ocean and ice are assumed to share the same grid. This means that the two models read in the same grid specification file. Even so, the domain decomposition on parallel systems may be different, and indeed they generally are due to different load balance issues between the two models.

Even though the ocean and ice models read the same grid specification file, they use the information from the grid file in a slightly different manner when setting up the respective model's tracer grid. In particular, the ocean model reads the tracer location directly from the arrays (x_T/geolon_t, y_T/geolat_t) written in the grid specification file. In contrast, the GFDL ice model reads (x_vert_T/geolon_vert_t, y_vert_T/geolat_vert_t) from the grid specifcation file and then averages these four vertex locatons to get the tracer location used in the ice model. The result is that diagnositics output from the two models have ocean and ice fields at slightly different locations for cases such as the tripolar grid when the grid is not spherical.

The ocean/ice grid specification file is generated by executing the ocean_grid_generator utility. The ocean_grid_generator utility generates the horizontal grid, vertical grid, and topography. A C-shell script is provided to compile relevant code to generate and run the executable to produce the grid file. To create the desired grid and topography, setting namelist options within the runscript is needed.

The horizontal grid can be conventional lon-lat spherical grid or a reprojected rotated tripolar grid (R. Murray, "Explicit generation of orthogonal grids for ocean models", 1996, J.Comp.Phys., v. 126, p. 251-273.). The choice is controlled by the namelist option "tripolar grid" (true for tripolar grid and false for lon-lat spherical grid). Note that cartesian beta-plane and f-plane geometries are set up within mom4, not within the grid generation preprocesing steps discussed here (see mom4/ocean_core/ocean_grids.F90 for beta-plane and f-plane namelist options).

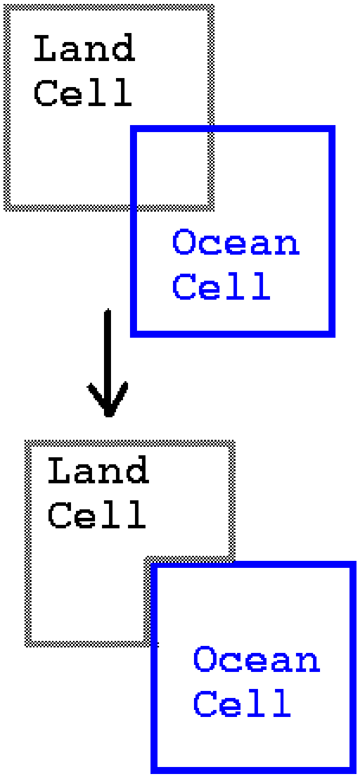
The grid_spec file contains the following horizontal grid information: geographic location of T,E,C and N-cell (Tracer, East, Corner, and North cells), half and full cell lengths (in meters), rotation information between logical (i.e., grid oriented) and geographic east of cell. The complete description of the horizontal grid and namelist option is available in hgrid [../src/preprocessing/generate_grids/ocean/hgrid.html]

The vertical grid information includes depth of tracer points and tracer_boundaries. The complete description of namelist option is available in vgrid [../src/preprocessing/generate_grids/ocean/vgrid.html]

The topography can be idealized (various examples are provided and others can be easily added through emulating those provided) or remapped from a source topography dataset. The type of topography is specified by the namelist variable "topography". Namelist "topog_depend_on_vgrid" specifies if the topography will depend on the vertical grid or not. To generate a grid for mom4, "topog_depend_on_vgrid" should always be true. A useful option for those upgrading older models to mom4 is "adjust_topo". If this option is set to false, there will be no adjustments made to the topography. See topog [../src/preprocessing/generate_grids/ocean/topog.html] for further details about topography namelist options.

5.3. The exchange grid for coupled models

"Exchange grid" information is required for coupled models (i.e., ocean/ice coupled to land and/or atmosphere) that employ the GFDL coupler technology. The exchange grid is defined by taking the union of the ocean/ice grid with the atmosphere and land grids. This union is then used to compute area integrals to allow for conservative mapping of fluxes between the component models.



The exchange grid information is generated by executing the make_xgrids utility. The execution of the make_xgrids utility will generate a netcdf file with the name grid_spec.nc. The grid_spec.nc contains the component model grids as well as the exchange grid information. In particular, the utility make_xgrids generates two exchange grids used by the FMS coupler: one grid for surface fluxes and another for runoff. **make_xgrids** is created by compiling its C source:

```
cc -O -o make_xgrids make_xgrids.c -I/usr/local/include -L/usr/local/lib
-lnetcdf -lm
```

creates the make_xgrids executable from C-source and the netCDF and standard math libraries. It is executed with the command

```
make_xgrids -o ocean_grid.nc -a atmos_grid.nc -l land_grid.nc
```

This execution produces a grid_spec.nc file (input files containing grid information for the ocean/sea-ice, atmosphere and land component models are indicated by the -o, -a and -l flags, respectively). The grid files ocean_grid.nc, atmosphere_grid.nc, and land_grid.nc all can be generated separately through the ocean_grid_generator utility. Normally at GFDL we select the same atmosphere and land model grid, but such is not necessary. When the land and atmosphere grids are the same, then we can reduce the execute command to

```
make_xgrids -o ocean_grid.nc -a atmos_grid.nc
```

If you further decide to choose same ocean, atmosphere and land grid, the execute command will be

```
make_xgrids -o ocean_grid.nc -a ocean_grid.nc
```

make_xgrids expects a netCDF format input specification for each of the component model grid files. For the ice/ocean grid (ocean_grid.nc), the following three fields are required:

1. wet - a 2D array of double precision numbers set to 1.0 where the ice and ocean models are active and 0.0 elsewhere. wet has im indices in the i-direction (pseudo east-west) and jm indices in the j-direction (pseudo north-south). These correspond to the size of the global arrays of temperature, salinity and ice thickness in the coupled climate model.

2. x_vert_T and y_vert_T - 3D double precision arrays (dimensioned im * jm * 4) that contain the longitudes and latitudes (respectively) of the four corners of T- cells. The numbers are in degrees.

For the netCDF format input specification for the atmosphere and land grid (atmos_grid.nc and/or land_grid.nc), x_vert_T and y_vert_t are required.

make_xgrids copies all fields of the ice/ocean grid specification file to its output file, grid_spec.nc, and then appends fields that specify the atmosphere and land model grids and then the surface and runoff exchange grids.

Using the Sutherland-Hodgeman polygon clipping algorithm (reference in next paragraph) for model cell interaction calculation, make_xgrids takes care that the land and ocean grids perfectly tile the sphere. The land model's domain is defined as that part of the sphere not covered by ocean (where wet=0 on the ocean grid). To accomplish this, the land cells must be modified to remove the ocean parts. This is done in make_xgrids by first taking the intersections of atmosphere and land cells. The overlap area between these cells and active ocean cells are then subtracted. Finally, the modified atmosphere/land intersections are aggregated into land cell areas and atmosphere/land exchange cell areas.

Model cell intersections are calculated using the Sutherland-Hodgeman polygon clipping algorithm (Sutherland, I. E. and G. W. Hodgeman, 1974: Reentrant polygon clipping, CACM, 17(1), 32-42.). This algorithm finds the intersection of a convex and arbitrary polygon by successively removing the portion of the latter that is "outside" each boundary of the former. It can be found in many computer graphics text books (e.g., Foley, J. D., A. van Dam, S. K. Feiner, and J. F. Hughes, 1990: Computer graphics: principles and practice, second edition. Addison Wesley, 1174 pp.). The implementation in make_xgrids is particularly simple because the clipping polygon is always a rectangle in longitude/latitude space. For the purpose of finding the line intersections in the clipping operations, the cell boundaries are assumed to be straight lines in longitude/latitude space. This treatment is only perfectly accurate for cells bounded by lines of longitude and latitude.

Spherical areas are calculated by taking the integral of the negative sine of latitude around the boundary of a polygon (Jones, P. W., 1999: First- and second-order conservative remapping schemes for grids in spherical coordinates. Monthly Weather Review, 127, 2204-2210.). The integration pathways are again straight lines in longitude/latitude space. make_xgrids checks that the sphere and the individual cells of the atmosphere and ocean grids are tiled by the surface exchange cells. The fractional tiling errors are reported.

5.4. Initial and Boundary Conditions

After generating the model grid, it is time to generate the initial and boundary conditions (ICs and BCs). These conditions are specific to the details of the model grid, so it is necessary to have the grid specificiation file in hand before moving to the IC and BC generation.

There are two options for ICs and BCs.

--Idealized Conditions. These conditions are based on subroutines that design idealized setups for either initial conditions (e.g., exponential temperature profile) or boundary conditions (e.g., cosine zonal wind stress). Code for these purposes is found in the idealized_ic and idealized_bc directories in the mom4 distribution. Details of available namelist choices are in the documentation file idealized_ic.html as well as the comments within the source code itself. Users can readily incorporate their favorite idealized IC or BC into the mom4 idealized preprocessing step by emulating the code provided.

--Realistic Conditions. These ICs and BCs generally result from a regridding routine to bring, say, the Levitus analysis onto the model grid for initializing a model, or for mapping surface fluxes onto the grid for boundary conditions. Code enabling the regridding functions is found in the preprocessing/regrid_2d, preprocessing/regrid_3d and preprocessing/regrid directories in the mom4 distribution.

In the remainder of this section, we detail code to generate the ICs and BCs of use for mom4.

5.4.1. 2d Regridding: the common approach

It is typical for air-sea fluxes of momentum, heat, and mosture to live on a grid distinct from the ocean model grid. In particular, most analyses are placed on a spherical latitude-longitude grid, whereas most global ocean models configured from mom4 are run with tripolar grids.

When running an ocean or ocean-ice model, it is useful to map the boundary fluxes onto the ocean model grid prior to the experiment. This preprocessing step saves computational time that would otherwise be needed if the fluxes were mapped each time step of a running experiment. To enable this regridding, one should access code in the preprocessing/regrid_2d directory. The original data must be on a latitude-longitude grid to use regrid_2d. The target/destination grid can be either latitude-longitude with arbitrary resolution, or tripolar with arbitrary resolution.

5.4.2. 2d Regridding: the less common approach

In some cases, one may wish to take a set of forcing fields from one tripolar mom4 experiment and regrid them onto another tripolar mom4 experiment with different grid resolution. In this case, it is necessary to regrid before running the experiment.

As of the mom4p0d distribution, there is a regridding tool within the preprocessing/regrid directory that enables one to regrid fields on one tripolar grid to another tripolar grid. Indeed, one can regrid source data from any logically rectangular grid (e.g., latitude-longitude grid or tripolar grid) to a target/destination grid that is any logically rectangular grid.

Note that this is new code, and so has been tested only for particular cases. So the user should be extra careful to scrutinize the results.

5.4.3. Setting the on_grid logical in the data_table

The "on_grid" logical in the data_table indicates whether an input file is on the grid of the model or not.

on_grid=.true. means that the input file is on the same grid as the ocean model. This is the recommended setting for models running with specified atmospheric forcing from data or an analysis product.

on_grid=.false. means the input file has data on a grid differing from the ocean model. This feature is allowed ONLY if the input data lives on a spherical grid. This is a relevant setting if one wishes to keep the input data on their native spherical grid. If the input data is non-spherical, then on_grid=.false. is NOT supported. Instead, it is necessary to preprocess the data onto the ocean model grid.

5.4.4. Regridding river runoff data

The tool preprocessing/runoff_regrid is of use to grid river runoff data onto the ocean model grid. In this case, runoff is moved to a nearest ocean/land boundary point on the new grid. Note that the source runoff dataset must be on a spherical latitude-longitude grid, whereas the target/destination grid can be spherical or tripolar. The regridding algorithm is conservative.

The conservative regridding scheme used in runoff_regrid is an area average scheme, which is similiar to the algorithm used in coupler flux exchange. If any land point has runoff data, after remapping runoff data onto destination grid, the runoff value of that land point will be moved to the nearest ocean point. Before using this tool, you must use make_xgrids to generate exchange grid information between the source grid and destination grid. The complete descritpion can be found in runoff_regrid.html.

5.4.5. Two ways to specify surface boundary fluxes

There are two ways to specify surface boundary fluxes when using the coupler feature of FMS. One is through flux exchange, and this employs a conservative algorithm as appropriate for running a coupled ocean-atmosphere model. It is assumed that the atmospheric model grid is spherical with arbitrary resolution. The other method is through data override, and this uses a non-conservative scheme. Data override is of use to selectively remove, say, one of the fluxes coming from an atmospheric model and replace this flux with that from data. GFDL modelers have found this feature to be very useful in diagnosing problems with a coupled model.

5.4.6. 3d Regridding for initial conditions or sponges

When generating realistic initial conditions for an ocean experiment, one generally requires the gridding of temperature and salinity, such as from the Levitus analysis product, onto the model's grid. For this purpose, we are in need of vertical grid information in addition to horizontal 2d information required for the surface boundary conditions. Hence, we use the preprocessing/regrid_3d. A similar procedure is required to develop sponge data.

The original data must be on a spherical grid in order to use regrid_3d. If the original data is on a tripolar grid, we should use preprocessing/regrid, which can map data from any logical rectangular grid onto any logical rectangular grid.

5.4.7. Comments on the regridding algorithms

For preprocessing/regrid_3d, preprocessing/regrid_2d and preprocessing/regrid, regridding is accomplished non-conservatively using a nearest neighbor distance weighting algorithm, or bilinear interpolation. The interpolation algorithm is controlled through the namelist option "interp_method".

Bilinear interpolation is recommanded for most cases since it provides a smooth interpolation when regridding from coarse grid to fine grid (the usual situation with model destination grids typically having resolution more refined than source data products), and it is more efficient. Efficiency can become a particularly important issue when developing initial and boundary conditions for a refined resolution model.

If the original data is on a tripolar grid, nearest neighbor distance weighting interpolation found in preprocessing/regrid must be used, since bilinear interpolation assumes the original data is on a latitude-longitude grid. For preprocessing/regrid_2d, preprocessing/regrid_3d and preprocessing/regrid using the nearest neighbor distance weighting algorithm, a maximum distance (in radians) can be selected using the namelist value max_dist. Namelist option "num_nbrs" can be adjusted for speed, although for most applications this refinement is not necessary.

The complete namelist description for these algorithms can be found in regrid_2d.html, regrid_3d.html and regrid.html.

5.4.8. Acceptable data formats

When the input data is on a latitude-longitude grid, preprocessing/regrid_2d and preprocessing/regrid_3d can be used.

When the input data is on a tripolar grid or a latitude-longitude grid, postprocessing/regrid can be used.

For sponge generation, acceptable input data sets must have NetCDF format with COARDS-compliance.

5.5. Time-related issues in forcing files

5.5.1. How it works and what to do if it fails

Previous versions of MOM used IEEE binary formats and MOM-specific headers to process forcing data. As of MOM4, data are stored in portable formats (NetCDF currently), and contain standardized metadata per the CF1.0 [http://www.cgd.ucar.edu/cms/eaton/cf-metadata/] convention.

Understading the functions of Fortran modules that handle metadata and time-related problems will be very helpful in identifying some user's problems. Some of the most frequently used modules are listed below:

mpp_io_mod	Low level I/O (open, close file, write, read,...)
axis_utils_mod	process metadata: identify cartesian axis information (X/Y/Z/T)
time_manager_mod	basic time operations, calendar, increment/decrement time
time_interp_mod	Computes a weight for linearly interpolating between two dates
time_interp_external_mod	top level routines for requesting data
data_override_mod	top level routines for requesting data

5.5.2. Test your forcing files before use

It is likely that you will encounter an error using "off-the-shelf" NetCDF files to force your ocean model. This could be due to inadequate metadata in the forcing files, mis-specification of the DataTable, or errors in the parsing of the axis information by axis_utils or get_cal_time. You'll need some tools to help you diagnose problems and apply the required fix.

The first thing you should do to setup a new forcing file is use the test program: **time_interp_external_mod:test_time_interp_external**. This test program calls time_interp_external at user-specified model times and returns information on how the times were decoded and the resulting interpolation indices and weights. It is STRONGLY suggested that you pass your forcing files through this program before including them in your model configuration. As you gain familiarity with the metadata requirements, you will more easily be able to identify errors and save a lot of time debugging.

The forcing test program is located in `src/preprocessing/test_time_interp_ext`. There is a csh version and a Perl version.

Compilation

```
mkmf -m Makefile -p test_time_interp_ext.exe -t $TEMPLATE -c -Dtest_time_interp_externa
```

running csh version

```
namelist options:
filename='foo.nc'                        ! name of forcing file
fieldname='foo'                          ! name of variable in file
year0=[integer]                          ! initial year to start model calendar
month0=[integer]                         ! initial month to start model calendar
day0=[integer]                           ! initial day to start model calendar
days_inc=[integer]                       ! increment interval for model calendar
ntime=[integer]                          ! number of model "timesteps"
cal_type=['julian','noleap','360day']  ! model calendar
```

running perl version

```
 test_time_interp_ext.pl -f 'foo.nc' -v 'foo' [--nt [int] --year0 [int] --month0 [int] -
```

Modifying the file metadata should hopefully prove straightforward. The NCO operators [http://nco.sourceforge.net] need to be installed on your platform. The utility "ncatted" is most useful for modifying or adding metadata. If for some reason, you are unable to install the NCO operators, you can use the NetCDF utilities "ncgen" and "ncdump" which come with the NetCDF package.

5.5.3. Common metadata problems

Can't identify cartesian axis information

axis_utils_mod:get_axis_cart should return the cartesian information. If this fails, you will get a somewhat cryptic error message: "file/fieldname could not recognize axis atts in time_interp_external". The best solution is to add the "cartesian_axis" attribute to the axes, e.g. "ncatted -a cartesian_axis,axis_name,c,c,"X" foo.nc".

Calendar attribute does not exist

This is a required attribute. time_manager_mod:get_cal_time converts time units appropriate to the specified calendar to the model time representation. If the "calendar" attribute does not exist, an error message appears "get_cal_time: calendar attribute required. Check your dataset to make sure calendar attribute exists " Use a ncatted command such as: "ncatted -a calendar,time_axis_name,c,c,"julian" foo.nc"

Currently, the FMS time_manager does not support the Gregorian calendar. So, for instance if you have forcing data that are encoded using the Gregorian calendar which has an average year length of 365.2425 days compared with the Julian calendar with an average year length of 365.25 days, assuming Julian calendar encoding will result in a drift of 0.75 days/100 years. If your forcing times are referenced to an early date such as "0001-01-01" your times will drift by 15 days by the year 2000. Until the Gregorian calendar is implemented in the FMS code, the recommended solution is to change the reference date in the forcing dataset using an application such as Ferret, click here [https://fms.gfdl.noaa.gov/mail/majordomo/archive.php?group_id=5&list_id=7&message_id=00753] to see the related discussion in the mom4p0 mailing list

6. Postprocessing regrid tool

6.1. Introduction

For many analysis applications, it is sufficient, and often preferable, to have output on the model's native grid (i.e., the grid used to run the simulation). Accurate computation of budgets, for example, must be done on the model's native grid, preferably online during integration. MOM4 provides numerous online diagnostics for this purpose.

Many applications, such as model comparison projects, require results on a common latitude-longitude spherical grid. Such facilitates the development of difference maps. For this purpose, we have developed a tool to regrid scalar and vector fields from a tripolar grid to a spherical grid. In principle, this tool can be used to regrid any logically rectangular gridded field onto a spherical grid. However, applications at GFDL have been limited to the tripolar to spherical regrid case.

In general, regridding is a difficult task to perform accurately and without producing noise or spurious results. The user should carefully examine regridding results for their physical integrity. Problems occur, in particular, with fields near mom4's partial bottom step topography in the presence of realistic topography and land/sea geometry. Indeed, we were unable to find a simple algorithm to handle regridding in the vertical that did not produce egregious levels of noise. Hence, the regridding tool provided with mom4 only handles horizontal regridding. The regridded data will thus be on the source vertical grid.

Model comparisons should ideally be performed only after regridding output using the same regridding algorithm. Unfortunately, such is not generally the case since there is no standard regridding algorithm used in the modeling community.

Please note that the regridding code is relatively new at GFDL. We greatly appreciate user's feedback.

6.2. How to use the regridding tool

The regridding algorithm provided with the mom4 distribution is located in the directory **postprocessing/regrid**

The algorithm accepts data from any logically rectangular grid (e.g., tripolar or latitude-longitude) and regrids to a spherical latitude-longitude grid. When the data is on the tracer cell (T-cell), the regridding interpolation is conservative. Thus, total heat, salt, and passive tracer remain the same on the two grids. However, when data is located at another position:

- corner or C-cell as for a B-grid horizontal velocity component
- east or E-cell as for an eastward tracer flux
- north or N-cell as for a northward tracer flux

then regridding is accomplished non-conservatively using a nearest neighbor distance weighting algorithm. It is for this reason that computationally accurate results are only available when working on the model's native grids.

The regridding tool reads grids information from a netcdf file, specified by the namelist "grid_spec_file". "grid_spec_file" contains source grid, destination grid and exchange grid information.

- source grid: src_grid.nc. This is the model's native grid. It results from running preprocessing grid generation code.
- destination grid: dst_grid.nc. This is the spherical latitude-lontitude grid. This grid can also be obtained from running preprocessing grid generation code. Be sure that the tripolar option is set to false to ensure that dst_grid.nc is spherical.
- exchange grid: grid_spec.nc. This is the union of the source grid and destination grid. The exchange grid is needed for conservative regridding. The same conservative regridding algorithm is used for coupled models with FMS. The tool to construct the exchange grid is know as "make_xgrids". It is located in the preprocessing directory. After grid_spec.nc is generated, it should be passed to the regrid tool through namelist "grid_spec_file"(No need to pass src_grid.nc and dst_grid.nc to the regrid tool).

To create the exchange grid, execute the command

make_xgrids -o src_grid.nc -a dst_grid.nc

The exchange grid creates a file grid_spec.nc. It has new fields with names:

AREA_ATMxOCN, DI_ATMxOCN, DJ_ATMxOCN, I_ATM_ATMxOCN, J_ATM_ATMxOCN, I_OCN_ATMxOCN, J_OCN_ATMxOCN, AREA_ATMxLND, DI_ATMxLND, DJ_ATMxLND, I_ATM_ATMxLND, J_ATM_ATMxLND, I_LND_ATMxLND, J_LND_ATMxLND, AREA_LNDxOCN, DI_LNDxOCN, DJ_LNDxOCN, I_LND_LNDxOCN, J_LND_LNDxOCN, I_OCN_LNDxOCN, J_OCN_LNDxOCN, xba, yba, xta, yta, AREA_ATM, xbl, ybl, xtl, ytl, AREA_LND, AREA_LND_CELL, xto, yto, AREA_OCN

It is critical that src_grid.nc DO NOT already have any of the above new exchange grid fields. If they do, then these fields should be removed using netcdf tools [http://nco.sourceforge.net/] such as ncks.

After the grid_spec.nc file is generated, it is passed into the regrid program through the nml option "grid_spec_file".

The regrid program reads model data from a netcdf file, which is specfied by the namelist variable "src_data". Again, src_data fields are gridded according to src_grid.nc. The number of fields to be regridded is specified by num_flds. The name of the fields (e.g., temp, salt) to be regridded is specified by the namelist variable "fld_name". Each field can be a scalar or vector. If a vector, then specify by vector_fld. Vector fields should always be paired together (e.g., u,v components to the horizontal current). The output file is a netcdf file specified by the namelist variable "dst_data".

The complete namelist option description is available in regrid.html or the code itself.

7. Preparing the runscript

7.1. The runscript

A runscript is provided in each test case [#mom4 test cases] directory (`mom4/exp/$test_case`) for each test case [#mom4 test cases]. Details can be found in quickstart_guide.html.

Incorporated in the FMS infrastructure is `MPP` [../src/shared/mpp/mpp.html] (Massively Parallel Processing), which provides a uniform message-passing API interface to the different message-passing libraries. If MPICH is installed, the user can compile the MOM4 source code with MPI. If the user does not have MPICH or the communications library, the MOM4 source code can be compiled without MPI by omitting the `CPPFLAGS` value `-Duse_libMPI` in the example runscript.

7.2. The diagnostics table

The diagnostics table allows users to specify the sampling rates and choose the output fields prior to executing the MOM4 source code. It is included in the `input` directory for each test case [#mom4 test cases] (`mom4/exp/$test_case/input`). A portion of a sample MOM4 diagnostic table is displayed below. Reference diag_manager.html [../src/shared/diag_manager/diag_manager.html] for detailed information on the use of diag_manager.

```
"Diagnostics for MOM4 test case"
1980 1 1 0 0 0
#output files
"ocean_month",1,"months",1,"hours","Time"
```

```
    "ocean_snap",1,"days",1,"hours","Time"
    #####diagnostic field entries####
    #==================================================================
    # ocean model grid quantities (static fields and so not time averaged))
    "ocean_model","geolon_t","geolon_t","ocean_month" "all",.false.,"none",2
    "ocean_model","geolat_t","geolat_t","ocean_month","all",.false.,"none",2
    #===================================================================
    # prognostic fields
    "ocean_model","temp","temp","ocean_month","all", "max", "none",2
    "ocean_model","age_global","age_global","ocean_month","all","min","none",2
    #===================================================================
    # diagnosing tracer transport
    "ocean_model","temp_xflux_sigma","temp_xflux_sigma","ocean_month","all",.true.,"none
    "ocean_model","temp_yflux_sigma","temp_yflux_sigma","ocean_month","all",.true.,"none
    #===================================================================
    # surface forcing
    "ocean_model","sfc_hflux","sfc_hflux","ocean_month","all",.true.,"none",2
    "ocean_model","sfc_hflux_adj","sfc_hflux_adj","ocean_month","all",.true.,"none",2
    #===================================================================
    # ice model fields
    "ice_model", "FRAZIL",   "FRAZIL",     "ice_month", "all", .true., "none", 2,
    "ice_model", "HI",    "HI",   "ice_month", "all", .true., "none", 2
    #------------------------------------------------------------------
```

The diagnostics manager module, `diag_manager_mod` [../src/shared/diag_manager/diag_manager.html], is a set of simple calls for parallel diagnostics on distributed systems. It provides a convenient set of interfaces for writing data to disk in NetCDF [http://www.unidata.ucar.edu/packages/netcdf/] format. The diagnostics manager is packaged with the MOM4 source code. The FMS diagnostic manager can handle scalar fields as well as arrays. For more information on the diagnostics manager, reference diag_manager.html [../src/shared/diag_manager/diag_manager.html].

7.3. The field table

The MOM4 field table is used to specify tracers and their advection schemes, cross-land tracer mixing, cross-land insertion, and other options. The field table is included in the runscript as a namelist and is written to an output file upon execution of the runscript.

```
"diag_tracers","ocean_mod","frazil"

file_in  = INPUT/ocean_frazil.res.nc
file_out = RESTART/ocean_frazil.res.nc
/

"prog_tracers","ocean_mod","temp"

horizontal-advection-scheme = quicker
vertical-advection-scheme = quicker
file_in  = INPUT/ocean_temp_salt.res.nc
file_out = RESTART/ocean_temp_salt.res.nc
/
```

```
"prog_tracers","ocean_mod","salt"

horizontal-advection-scheme = mdfl_sweby
vertical-advection-scheme = mdfl_sweby
file_in  = INPUT/ocean_temp_salt.res.nc
file_out = RESTART/ocean_temp_salt.res.nc
/



"tracer_packages","ocean_mod","ocean_age_tracer"

names = global
horizontal-advection-scheme = mdfl_sweby
vertical-advection-scheme = mdfl_sweby
file_in  = INPUT/ocean_age.res.nc
file_out = RESTART/ocean_age.res.nc
min_tracer_limit=0.0
/



"namelists","ocean_mod","ocean_age_tracer/global"

slat = -90.0
nlat =  90.0
wlon =   0.0
elon = 360.0
/



"xland_mix","ocean_mod","xland_mix"
"xland","Gibraltar","ixland_1=274,ixland_2=276,jxland_1=146,jxland_2=146,kxland_1=1,kxla
"xland","Gibraltar","ixland_1=274,ixland_2=276,jxland_1=147,jxland_2=147,kxland_1=1,kxla
"xland","Black-Med","ixland_1=305,ixland_2=309,jxland_1=151,jxland_2=152,kxland_1=1,kxla
"xland","Black-Med","ixland_1=306,ixland_2=309,jxland_1=151,jxland_2=153,kxland_1=1,kxla

"xland_insert","ocean_mod","xland_insert"
"xland","Gibraltar","ixland_1=274,ixland_2=276,jxland_1=146,jxland_2=146,kxland_1=1,kxla
"xland","Gibraltar","ixland_1=274,ixland_2=276,jxland_1=147,jxland_2=147,kxland_1=1,kxla
"xland","Black-Med","ixland_1=305,ixland_2=309,jxland_1=151,jxland_2=152,kxland_1=1,kxla
"xland","Black-Med","ixland_1=306,ixland_2=309,jxland_1=151,jxland_2=153,kxland_1=1,kxla


"diff_cbt_enhance","ocean_mod","diff_cbt_enhance"
"diffcbt","Gibraltar","itable=274,jtable=146,ktable_1=1,ktable_2=18,diff_cbt_table=0.01"
"diffcbt","Gibraltar","itable=276,jtable=146,ktable_1=1,ktable_2=18,diff_cbt_table=0.01"
"diffcbt","Gibraltar","itable=274,jtable=147,ktable_1=1,ktable_2=18,diff_cbt_table=0.01"
"diffcbt","Gibraltar","itable=276,jtable=147,ktable_1=1,ktable_2=18,diff_cbt_table=0.01"
"diffcbt","Black-Med","itable=305,jtable=151,ktable_1=1,ktable_2=6,diff_cbt_table=0.01"
"diffcbt","Black-Med","itable=309,jtable=152,ktable_1=1,ktable_2=6,diff_cbt_table=0.01"
"diffcbt","Black-Med","itable=306,jtable=151,ktable_1=1,ktable_2=6,diff_cbt_table=0.01"
"diffcbt","Black-Med","itable=309,jtable=153,ktable_1=1,ktable_2=6,diff_cbt_table=0.01"/
```

In the first section of the field table, the user can specify tracers to be used in the simulation. Although there is no limit to the number of tracers specified, temperature (`temp`) and salinity (`salt`) must be included. The user may also define the horizontal and vertical tracer advection schemes. For more information on the field manager, reference field_manager.html [../src/shared/field_manager/field_manager.html].

In climate modeling, it is often necessary to allow water masses that are separated by land to exchange tracer and surface height properties. This situation arises in models when the grid mesh is too coarse to resolve narrow passageways that in reality provide crucial connections between water masses. The cross-land mixing and cross-land insertion establishes communication between bodies of water separated by land. The communication consists of mixing tracers and volume between non-adjacent water columns. Momentum is not mixed. The scheme conserves total tracer content, total volume, and maintains compatibility between the tracer and volume budgets. The grid points where this exchange takes place, and the rates of the exchange, are specified in the field table.

For some cases, it is necessary to set a large vertical tracer diffusivity at a specified point in the model, say next to a river mouth to ensure fresh water is mixed vertically. These diffusivities are specified in the field table.

For a technical description of cross-land tracer mixing and insertion, please reference A Technical Guide to MOM4 [http://www.gfdl.noaa.gov/~smg/pointers/geo_physics_abstracts/guide.pdf].

7.4. mppnccombine

Running the MOM4 source code in a parallel processing environment will produce one output NetCDF [http://www.unidata.ucar.edu/packages/netcdf/] diagnostic file per processor. `mppnccombine` [../postprocessing/mppnccombine.c] joins together an arbitrary number of data files containing chunks of a decomposed domain into a unified NetCDF [http://www.unidata.ucar.edu/packages/netcdf/] file. If the user is running the source code on one processor, the domain is not decomposed and there is only one data file. `mppnccombine` [../postprocessing/mppnccombine.c] will still copy the full contents of the data file, but this is inefficient and `mppnccombine` [../postprocessing/mppnccombine.c] should not be used in this case. Executing `mppnccombine` [../postprocessing/mppnccombine.c] is automated through the runscripts [#runscript]. The data files are NetCDF [http://www.unidata.ucar.edu/packages/netcdf/] format for now, but IEEE binary may be supported in the future.

`mppnccombine` [../postprocessing/mppnccombine.c] requires decomposed dimensions in each file to have a `domain_decomposition` attribute. This attribute contains four integer values: starting value of the entire non-decomposed dimension range (usually 1), ending value of the entire non-decomposed dimension range, starting value of the current chunk's dimension range and ending value of the current chunk's dimension range. `mppnccombine` also requires that each file have a `NumFilesInSet` global attribute which contains a single integer value representing the total number of chunks (i.e., files) to combine.

The syntax of `mppnccombine` [../postprocessing/mppnccombine.c] is:

```
mppnccombine [-v] [-a] [-r] output.nc [input ...]
```

Table 1. mppnccombine arguments

-v	print some progress information
-a	append to an existing NetCDF [http://www.unidata.ucar.edu/packages/netcdf/] file
-r	remove the '.####' decomposed files after a successful\ run

An output file must be specified and it is assumed to be the first filename argument. If the output file already exists, then it will not be modified unless the option is chosen to append to it. If no input files are specified, their names will be based on the name of the output file plus the extensions '.0000', '.0001', etc. If input files are specified, they are assumed to be absolute filenames. A value of 0 is returned if execution is completed successfully and a value of 1 indicates otherwise.

The source of `mppnccombine` [../postprocessing/mppnccombine.c] is packaged with the MOM4 module in the `postprocessing` directory. `mppnccombine.c` should be compiled on the platform where the user intends to run the FMS MOM4 source code so the runscript [#runscript] can call it. A C compiler and NetCDF [http://www.unidata.ucar.edu/packages/netcdf/] library are required for compiling `mppnccombine.c`:

```
cc -O -o mppnccombine -I/usr/local/include -L/usr/local/lib mppnccombine.c -lnetc
```

8. Examining the output

8.1. Sample model output

Sample MOM4 model output data files are available to registered [#source code and data sets] MOM4 users on GFDL's NOMADS server. The output data are organized into directories that bear the same names as the test cases [#mom4 test cases]. For example, output for test case `test5` can be found in directory `test5`. Output files are classified into three subdirectories:

- `ascii`: the description of the setup of the run and verbose comments printed out during the run.
- `restart`: the model fields necessary to initialize future runs of the model.
- `history`: output of the model, both averaged over specified time intervals and snapshots.

Note that these output files are compressed using **tar**. All `.tar` files should be decompressed for viewing. The decompress command is:

```
tar -xvf filename.tar
```

8.2. Analysis tools

There are several graphical packages available to display the model output. These packages vary widely depending on factors, such as the number of dimensions, the amount and complexity of options available and the output data format. The data will first have to be put into a common format that all the packages can read. FMS requires the data to be stored in NetCDF [http://www.unidata.ucar.edu/packages/netcdf/] format since it is so widely supported for scientific visualization. The graphical package is also dependent upon the computing environment. For ocean modeling, ncview [http://meteora.ucsd.edu/~pierce/ncview_home_page.html], Ferret [http://ferret.wrc.noaa.gov/Ferret/] and GrADS [http://grads.iges.org/grads/] are most commonly used.

9. Running mom4 on various computational platforms

An often frustrating aspect of numerical modeling is getting the computational environment set up so the model compiles, executes, completes, and saves data in a smooth and dependable manner. There are numerous platform configurations, compiler variations, data storage capabilities, and experimental designs that make it likely that each user will encounter unique issues.

At GFDL, we try to develop code that is portable and readily usable on many platforms. Nonetheless, given the multitude of configurations, we cannot anticipate all problems. So users should expect to spend some time setting up the model's computational environment when downloading the code for the first time.

We rely on users within the mom4 community to provide feedback regarding how they got the model to run on their respective platforms. We greatly appreciate when users send their "progress report" to the mom4 email list. We solicit information on issues that may make future releases less problematic. We also appreciate your detailed documentation that may approach "hand holding" to better allow new users to successfully run the model. Indeed, imagine being a fresh graduate student or postdoc, only just recently learning Unix and Fortran, aiming to download and run mom4 on their PC! The more experienced users share knowledge in a digested and organized manner, the easier it will be for others to realize success with the code.

The purpose of this section is to organize some of the many emails that have been sent to the mom4 email list that focus on computational platform issues. Given limited resources and limited access to platforms outside GFDL, we have not edited these emails. Users who wish to update or edit information provided in an earlier email please feel free to do so.

NOTE: Some platform related problems may have already been solved in the latest release. Please check the section "Code Updates" for release note of each release.

9.1. IBM

```
Compiler warning in mpp_domains_comm.F90
https://fms.gfdl.noaa.gov/mail/majordomo/archive.php?group_id=5&list_id=7&message_id=006
https://fms.gfdl.noaa.gov/mail/majordomo/archive.php?group_id=5&list_id=7&message_id=006
https://fms.gfdl.noaa.gov/mail/majordomo/archive.php?group_id=5&list_id=7&message_id=006
https://fms.gfdl.noaa.gov/mail/majordomo/archive.php?group_id=5&list_id=7&message_id=006
https://fms.gfdl.noaa.gov/mail/majordomo/archive.php?group_id=5&list_id=7&message_id=007
https://fms.gfdl.noaa.gov/mail/majordomo/archive.php?group_id=5&list_id=7&message_id=007
https://fms.gfdl.noaa.gov/mail/majordomo/archive.php?group_id=5&list_id=7&message_id=007
MPI buffer problem
https://fms.gfdl.noaa.gov/mail/majordomo/archive.php?group_id=5&list_id=7&message_id=008
https://fms.gfdl.noaa.gov/mail/majordomo/archive.php?group_id=5&list_id=7&message_id=008
https://fms.gfdl.noaa.gov/mail/majordomo/archive.php?group_id=5&list_id=7&message_id=008
```

9.2. Intel IFC on Beowulf

```
Mom4 test4 can run on 1 processor on Beowulf machine
https://fms.gfdl.noaa.gov/mail/majordomo/archive.php?group_id=5&list_id=7&message_id=004
```

9.3. PGI

```
Mom4p0c run on PGI5.2-4
https://fms.gfdl.noaa.gov/mail/majordomo/archive.php?group_id=5&list_id=7&message_id=006
https://fms.gfdl.noaa.gov/mail/majordomo/archive.php?group_id=5&list_id=7&message_id=006
https://fms.gfdl.noaa.gov/mail/majordomo/archive.php?group_id=5&list_id=7&message_id=007
```

9.4. SUN

```
Running mom4 on a Sun v60x (intel Xeon) cluster
https://fms.gfdl.noaa.gov/mail/majordomo/archive.php?group_id=5&list_id=7&message_id=006
```

9.5. NEC

```
MOM4p0 Length of error messages on NEC SX platforms
https://fms.gfdl.noaa.gov/mail/majordomo/archive.php?group_id=5&list_id=7&message_id=006
```

```
Mom4 test1 on NEC SX6i
https://fms.gfdl.noaa.gov/mail/majordomo/archive.php?group_id=5&list_id=7&message_id=010
```

10. Code updates

MOM4 is an evolving code. As features are added or removed, bugs are resolved, and documentation clarified, we aim to provide the user community with access to our latest code used in-house at GFDL. The purpose of this section is to summarize the main features of each model version.

For new users, please download the most recent release. For those having used mom4 for sometime, your need to update should be based on a balance between accessing new features or bug fixes versus the overhead needed to update your respective model experiments. It is recommended that you access new versions of mom4 only after completion of a particular line of research, rather than updating in the middle. Notably, we do not ensure that each update is fully backwards compatible.

10.1. MOM4p0b updates

This is a list of files modified in March/April 2004 for the MOM4p0b release of MOM4.

shared/exchange/xgrid.f90 [../src/shared/exchange/xgrid.html]	Minor edits to make the code compliant with IFC version 8.
ice_param/ocean_albedo.f90 [../src/ice_param/ocean_albedo.html]	Minor edits to make the code compliant with IFC version 8.
shared/column_diagnostics/column_diagnostics.f90 [../src/shared/column_diagnostics/column_diagnostics.html]	Minor edits to make the code compliant with IFC version 8.
shared/horiz_interp/horiz_interp.f90 [../src/shared/horiz_interp/horiz_interp.html]	Change the step size of radial search from 1 to 100. This will improve the efficiency and a workaround to fix a bug.
preprocessing/generate_grids/ocean/topog.f90 [../src/preprocessing/generate_grids/ocean/topog.html]	Modify io format to fix one error message.
preprocessing/generate_grids/ocean/ocean_grid_generator.csh [../src/preprocessing/generate_grids/ocean/ocean_grid_generator.html]	Minor comments update
shared/mpp/mpp_io.f90 [../src/shared/mpp/mpp_io.html]	Added optional argument "time_method" to write_meta_field subroutine. This is used by diag_manager to document how a diagnostic is manipulated along the time axis, e.g. averaging, max, min. The metadata used are compliant with CF1.0 metadata standard.
shared/mpp/mpp_read_2Ddecomp.h [../src/shared/mpp/mpp_read_2Ddecomp.h]	Bug fix for lenx and leny variables
shared/diag_manager [../src/shared/diag_manager/diag_manager.html]	In addition to time average now users can have min, max in the same time period.

coupler/coupler_main.f90 [../src/coupler/coupler_main.html]	Replace constant number 2 with parameter variable FATAL in error message.
drivers/ocean_solo.F90 [../src/mom4/drivers/ocean_solo.html]	Remove redundant mpp_clock for Ocean.
postprocessing/regrid [../postprocessing/regrid/regrid.html]	using netcdf library to read and write data, instead of using mpp_io_mod. Added vertical interpolation. Add namelist do_laplace_extrap to improve efficiency when doing vertical interpolation.
ocean_core/ocean_domains.F90 [../src/mom4/ocean_core/ocean_domains.html]	Added (len=*) for variable name in subroutine set_ocean_domain.
ocean_core/ocean_velocity_advect.F90 [../src/mom4/ocean_core/ocean_velocity_advect.html].	Corrected bug identified by Pacanowski. For bottom partial cells with k=nk, the vertical advection of horizontal velocity was a factor of 2 too large. A mask has been added to correct this bug.
ocean_core/ocean_grids.F90 [../src/mom4/ocean_core/ocean_grids.html]	Send to diagnostics for area_u had id_area_t in send_data, trivial bug.
ocean_core/ocean_topog.F90 [../src/mom4/ocean_core/ocean_topog.html]	Added nml for flat_bottom for use in helping to debug problems related to bottom topography.
ocean_core/ocean_freesurf.F90 [../src/mom4/ocean_core/ocean_freesurf.html]	Added nml option to have eta_t = 0.0 at the initial condition. This is useful when have restart files that have nonzero eta, and wish to debug without changing the restart files. Added nml option check_volume_conserve for debugging purposes.
ocean_param/mixing/vert/kpp/ocean_vert_mix_coeff.F90 [../src/mom4/ocean_param/mixing/vert/kpp/ocean_vert_mix_coeff.html]	Removed the tidal_mix option as this remains a topic of research.
ocean_param/mixing/neutral/ocean_neutral_physics.F90 [../src/mom4/ocean_param/mixing/neutral/ocean_neutral_physics.html]	Unified the calculation of the surface boundary layer. Now include information about the KPP boundary layer in the computation of the "neutral physics boundary layer".
ocean_tracers/ocmip2_biotic.F90 [../src/mom4/ocean_tracers/ocmip2_biotic.html]	Corrected bug identified by John Dunne. Bug fix necessary to keep model from bombing.
ocean_param/sources/sponge/ocean_sponges.F90 [../src/mom4/ocean_param/sources/sponge/ocean_sponges.html]	Removed code that inverted the input sponge coefficients. The input sponge_coeff NetCDF file should now give these coefficients in inverse seconds, rather than in seconds. Reasoning: (1) matches the documentation; (2) more intuitively linked to the name "sponge_coeff," with a large coeff implying a strong sponge; and (3) allows tapering of sponge coefficients to exactly zero.
shared/mpp_domains.F90 [../src/shared/mpp/mpp_domains.html]	Bug fix for C grid case, does not affect MOM4 since MOM4 runs on B grid.
shared/mpp_update_domains2D.h [../src/shared/mpp/mpp_update_domains2D.h]	Bug fix for C grid case, does not affect MOM4 since MOM4 runs on B grid.

10.2. MOM4p0c updates

The following lists some new features of the MOM4p0c release in August 2004.

Warning

- MOM4p0c will NOT bitwise reproduce earlier MOM4 releases. However, major changes in answers beyond roundoff differences should not be expected when running in an analogous configuration.
- mom4p0c code REQUIRES the use of mom4p0c datasets to avoid fatal error.

New or modified algorithms available in MOM4p0c

- The time tendency for tracer and baroclinic velocity can be discretized two ways. (1) The first approach uses the traditional leap-frog method along with a Robert-Asselin time filter. (2) MOM4p0c provides an additional method, which is strongly recommended. Here, the time tendency is discretized with a two-level forward step, which eliminates the need to time filter. Tracer and velocity are staggered in time, thus providing second order accuracy in time. For certain model configurations, this scheme is roughly twice as stable as the leap-frog, thus allowing for a factor of two in computational savings. Without the time filtering needed with the leap-frog, the new scheme conserves total tracer to within numerical roundoff. Extensive tests indicate that the simulations are quite similar qualitatively and quantitatively to those run with the leap-frog scheme. However, the user is encouraged to verify such prior to moving forward with the new time stepping scheme. Note that the new scheme shares much in common with time stepping used in the Hallberg Isopycnal Model and the MIT GCM. This new scheme is the default in MOM4p0c.
- The barotropic mode can be updated using either the older leap-frog method, or a newer defaulted predictor-corrector method motivated by a similar method used in the Hallberg Isopycnal Model.
- Thickness weighted tracer concentration and thickness weighted baroclinic velocity are updated, rather than tracer concentration and velocity. This new approach provides simpler conservation budgets, in which there is no longer a source-like term associated with the time tendency of the surface height.
- Optimizations have been implemented to reduce array syntax and to improve parallelization. This work is ongoing, with assistance from the MOM4 user community greatly appreciated.
- A new derived type has been introduced: ocean_thickness_type. This type absorbs elements of the ocean_grid_type that invove time dependent thicknesses of layers. It is anticipated that this type will expand in the future as MOM4 evolves towards generalized vertical coordinates. This change required interfaces changes.
- The Time derived type has been removed from the other derived types in order to simplify the definitions of the types. This change required interfaces changes.
- The non-Boussinesq option has been jettisoned in order to simplify some of the code. It is anticipated that the next release of MOM4 will be MOM4p1 (late 2004 or early 2005), and this code will include a pressure-coordinate option to allow for simpler implementation of non-Boussinesq kinematics/dynamics.

Corrected algorithms in MOM4p0c

- The shortwave fraction computed in `mom4/ocean_param/sources/shortwave/ocean_shortwave_pen.F90` [../ src/ mom4/ ocean_param/sources/shortwave/ocean_shortwave_pen.html] was not constrained to monotonically decrease with depth. In some cases, the fraction could spuriously increase. A new nml logical has been introduced: enforce_sw_frac. The default of this logical is true.
- Tidal forcing in `mom4/ocean_core/ocean_freesurf.F90` [../ src/ mom4/ ocean_core/ ocean_freesurf.html] has been corrected. The older code had problems with the phases.

New features of the FMS infrastructure

- Extensive work has been included to enhance the portability and flexibility of FMS codes. This work is ongoing, with assistance from the MOM4 user community greatly appreciated.
- The new version of `shared/time_manager.f90` [../src/shared/time_manager/time_manager.html] requires that every dataset with a time dimension also have a calendar attribute associated with this time. This new attribute is now required because problems were found with the old time manager when running the model assuming a repeating annual cycle using the NOLEAP calendar type, yet without the datasets also having this calendar type. A spurious shift in the date accumulated over time, which caused shifts in the forcing relative to what the model intended. The calendar attribute is missing from many datasets released with mom4p0a and mom4p0b. Since the missing of calendar attribute would cause a fatal error in mom4p0c release, NEW datasets that have calendar attribute are required for mom4p0c release.
- Because getting NetCDF datasets in a proper format for FMS code remains a frustrating aspect of running MOM4, we have provided a new stand-alone utility that tests a data file for compatibility with FMS. This

utility is in `shared/time_interp_external.F90` [../ src/ shared/ time_interp/ time_interp_external.html] and there is an associated run script at src/preprocessing/test_time_interp_ext that the user should modify for their machines. Use of this utility to debug the datasets prior to running MOM4 can help to remove a significant level of frustration associated with building a new model with new datasets.

- Another substantial update to the FMS infrastructure involves the mpp code, such as `shared/mpp.F90` [../src/shared/mpp/mpp.html]. This code has been significantly reorganized for easier understanding, and many new features have been added. One new feature that has been found quite useful is that it is now possible to update a set of fields across processors after accumulating the full set, rather than sending each member individually. In general, sending one very large package of data for mpp_update_domains is far more efficient than sending multiple smaller packages. This new feature is most useful, for example, when sending multiple tracers or tracer tendencies for updates in modules such as ocean_neutral_physics.F90 and tracer advection. (Note that only the Sweby advection scheme has been modified to allow for this improved efficiency. Other schemes can be modified similarly. Since they are not commonly used at GFDL, they have not been modified).

10.3. MOM4p0d updates

- mom4p0d code employs the same data files as mom4p0c
- Updates for mom4p0d are minimal within the mom4 portion of the release. The main emphasis is with the associated code related to preprocessing, postprocessing, and other elements of FMS. The mom4p0d release is based on the internal GFDL release of FMS known as "Lima."
- In mom4p0d release the Frequently Asked Questions (FAQ) have been extensively rewritten. The new FAQ [http://www.gfdl.noaa.gov/~gtn/mom4_faq.html] contains many questions that have been discussed in the mom4p0 mailing list.

The following lists some new features of the MOM4p0d release in May 2005.

mom4 code

- ocean_param/mixing/horz/velocity/bih/general/ocean_bih_friction.F90
  - bug fix for biharmonic general friction in computation of the viscosity will CHANGE answer in test2
- ocean_core/ocean_obc.f90
  - OBC now runs with static memory allocation
- ocean_core/ocean_freesurf.F90
  - always initialize the tidal forcing, as tidal_forcing_init allocates some arrays which are used with the dynamical allocation model.
- mom4/ocean_core/ocean_grids.F90
  - add namelist option beta_plane and f_plane in ocean_grids_nml. With default value false.

  - fix bug that some diagnostics data axis is not in correct location.
- mom4/ocean_core/ocean_tracer.F90
  - fix the +0 and -0 restart reproducing problem for frazil in SGI/Irix.
- ocean_diag/ocean_tracer_diag.F90

  ocean_diag/ocean_velocity_diag.F90

  mom4/ocean_param/sources/overflow/ocean_overflow.F90

  - add namelist do_bitwise_exact_sum to the diagnostic module. Set true to do bitwise exact global sum. When it is false, the global sum will be non-bitwise-exact, but will significantly increase efficiency. The default value is false.

- ocean_core/ocean_types.F90
  - convert pointer to allocatable array components in data type to improve effieciency on Altix machine when running in dynamic case.
- ocean_param/mixing/neutral/ocean_neutral_physics.F90
  - removed unnecessary memory allocation when neutral physics is not used.

  - fixed incorrect index.
  - corrected update_domain for ustar and vstar, will not change answer, only affect diagnostics
- ocean_diag/ocean_tracer_util.F90

  ocean_diag/ocean_adv_vel_diag.F90
  - added Fortran intrinsic functions "maxloc" and "minloc" in a new algorithm for computing tracer_min_max and/or maximum CFL number. These intrinsic functions allow for more vectorized algorithms, so should see some efficiency improvement especially on vector machines.
- mom4/ocean_param/sources/rivermix/ocean_riverspread.f90
  -added "if(nspread==0) return" in spread_river_horz routine. Change the interface of spread_river_horz routine to avoid a possible aliasing.
- mom4/ocean_core/ocean_sbc.F90
  Compatible with the interface change of routine spread_river_horz in ocean_riverspread.f90.
-

```
mom4/ocean_param/sources/shortwave/ocean_shortwave_pen.F90
mom4/ocean_param/sources/xlandinsert/ocean_xlandinsert.f90
mom4/ocean_param/sources/xlandmix/ocean_xlandmix.f90
mom4/ocean_param/mixing/polarfilter/ocean_polar_filter.f90
mom4/ocean_param/mixing/sigma/ocean_sigma_diffuse.F90
mom4/ocean_param/mixing/horz/tracer/bih/ocean_horz_diffuse.f90
mom4/ocean_param/mixing/horz/tracer/lap/ocean_horz_diffuse.f90
mom4/ocean_param/mixing/horz/velocity/bih/const/ocean_bih_friction.f90
mom4/ocean_param/mixing/horz/velocity/bih/general/ocean_bih_friction.F90
mom4/ocean_param/mixing/horz/velocity/lap/const/ocean_lap_friction.f90
mom4/ocean_param/mixing/horz/velocity/lap/general/ocean_lap_friction.F90
mom4/ocean_param/mixing/vert/ocean_vert_mix.f90
mom4/ocean_param/mixing/vert/kpp/ocean_vert_mix_coeff.F90
```

  - removed unnecessary memory allocation if the corresponding module's namelist is false

shared code

- A few minor changes have been made to MPP. Some have to do with extending shared memory functionality within MPP. This includes a new file threadloc.c.
- The file shared/mpp/include/os.h has been renamed and relocated to the top level include/fms_platform.h
- shared/diag_manager/diag_manager.f90

  - add standard_name to output fields.

  - eliminate the need to pass Time to send_data for static fields

  - add code that writes the meta data associated with register fields to an ascii file called "diag_field_log.out".
- shared/mpp/mpp_parameter.F90 and shared/mpp/mpp_domains_comm.F90
  -use Z instead of X for hexadecimal data

- shared/axis_utils/axis_utils.F90
  add a test program test_axis_utils. Remove unnecessary input grid to avoid bad data in the input data. It shouldn't change solution for any present experiments.
- shared/time_interp/time_interp_external.F90
  Include filename/fieldname in error message of init_external_field.
- shared/mpp/mpp_io_util.F90
  bugfix in subroutine mpp_get_axes().
- shared/horiz_interp/horiz_interp_spherical.f90
  add a namelist "search_method" to horiz_interp_spherical_nml. The reason is that the radial search may be not quite accurate in some cases. In this situation, you can always set search_method to "full_search", which will be always accurate but less efficiency comparing to "radial_search". "radial_search" is the default value. In most cases, "radial_search" and "full_search" will produce same results other than order of operation. With "radial_search", it is important to choose suitable max_dist (an optional argument in horiz_interp_spherical_init).
- shared/mpp/mpp_io_read.F90
  change to fix some possible +0/-0 restart problem.

bin/mkmf, bin/mkmf.template.ibm

- update for IBM platform

coupler/flux_exchange.f90

- By default, the global exchange grid u_star will not be interpolated from atmospheric grid. This is different from mom4p0c behavior and will change answers. To reproduce old answers, use

&flux_exchange_nml ex_u_star_smooth_bug = .true.

preprocessing

- preprocessing/mom4_prep/idealized_ic/idealized_ic.f90
  -change the attribute cartesian_axis of zt from "z" to "Z".
- preprocessing/mom4_prep/idealized_bc/idealized_bc.f90
  -fix the bug that variable yu1 is not initialized for option cosine_winds.
- preprocessing/generate_grids/make_xgrids/make_xgrids.c
  - remove the hardwired path to netcdf.h and update the compiling command
- preprocessing/generate_grids/ocean/topog.f90
  - change the default value of nml interp_method from "spherical" to "bilinear". Some code clean-up.
- preprocessing/regrid
  This tools is based on postprocessing/regrid. It can regrid data from 2-D/3-D logical rectangular grid onto logical rectangular grid. The interpolation will be non-conservative. Some tests shown that both forcing data sets of mom4 test case restart files from mom4 test run and can be used as input source data for regridding. There is , however, no guarantee for all the situations.
- preprocessing/regrid_2d
  Some adjustments e.g. laplacian extrapolation will be done only when it is needed.
- preprocessing/regrid_3d
  Replace nanmelist do_vertical_interp with use_source_vertical_grid. When use_source_vertical_grid is true, no vertical interpolation will be done. Otherwise vertical interpolation will be done if the source and destination grid have different vertical grid. Replace apply_dest_mask with apply_mask to add the option to allow source data mask to determine destination data mask. Redesign the code to decrease the memory usage.

postprocessing/regrid

Redesign this regrid tool. Change the interpolation algorithm for tracer field. Tracer field will be remapped using conservative scheme. The regridding will be limited from any logical rectangular grid (tripolar of latitude-longitude grid) to any latitude-longitude grid. Removed namelist do_vertical_interp. This tool will not do vertical interpolation. The reason is that the linear vertical interpolation will create some unwanted noises.

10.4. MOM4p0d_patch20050717 updates

The following lists some new features of the MOM4p0d_patch20050717 release in July 2005.

- - bin/mkmf.template.ibm

  fixed the problem on IBM machine that mld_id is not defined.

- - src/mom4/ocean_diag/ocean_adv_vel_diag.F90

  bug fix, reported by Tricia Balle.

- - mom4/ocean_core/ocean_freesurf.F90

  Fix bug in tidal_forcing_init. When calculating coslat2 and sin2lat, using yt instead of using xt.

- - mom4/ocean_param/mixing/neutral/ocean_neutral_physics.F90

  The array sumz had an unnecessary extra dimension of size num_prog_tracers. This dimension has been removed. No answers change due to this modification.

- - mom4/ocean_core/ocean_thickness.F90

  When register field dhu, the axis Grid%tracer_axes is replaced by Grid%vel_axes_uv.

- - shared/horiz_interp/horiz_interp.f90

  - shared/horiz_interp/horiz_interp.html

  added documentation for argument grid_at_center of interface horiz_interp_init.

- - shared/horiz_interp/horiz_interp_bilinear.f90

  deal with the situation that the starting longitude of source grid is greater than 0.

- - shared/mpp/mpp_domains_comm.F90

  - shared/mpp/include/mpp_do_global_field_old.h

  fixed the bug that will cause problem when doing x or y-direction global field.

- - shared/mpp/include/mpp_do_updateV_old.h

  fixed a harmless bug. corrected the receive size for the BGRID_NE case.

- - shared/mpp/include/system_clock.h

  added system_clock_default for case where -Duse_libMPI is not used and compilation not on SGI.

- - shared/mpp/threadloc.c

  - include/fms_platform.h

  removed if_defined(_IBMC_) to make it IBM compatible.

- - preprocessing/regrid_3d/regrid_3d.f90 added namelist ntimes_saved and timelevel_saved to add the capability to output selective time levels. The main purpose is to allow the program to generate one or two time level initial condition for mom4 when the source data has more than 2 time levels.

- - ice_sis/ice_model.f90 **will CHANGE results**

  fixed the rotation bug and call cut_check only when the grid is tripolar grid. A namelist cm2_bugs (defined in ice_type.f90) is introduced. The default value false represents correct rotation. In order to reproduce old results, need to set cm2_bugs to true.

- - ice_sis/ice_type.f90

  added namelist cm2_bugs to fix the stress rotation bug. When cm2_bugs is true, will use bug version stress rotation and when cm2_bugs is false, will use corrected stress rotation. The default value is false.

- - ice_sis/ice_grid.f90

  read boundary condition from grid file, remove the tripolar y-boundary condition and cyclic x-boundary condition assumption. Replace netcdf routine with mpp_io and fms_io routine.

- - shared/data_override/data_override.F90

  updated the code to have the option to select between "bilinear" and "bicubic" interpolation if the data is not on the model grid. In order to use this feature, some change in data_override_table is needed. Instead using logical variable to indicate on grid or not, the fifth entry need to be replaced by string variable to indicate interpolation method. The string value can be "bilinear", "bicubic", "default" (same as "bilinear") and "none" (no interpolation needed, the data is on model grid). In a single data_override table, only one format of entry is allowed (either the fifth entry of all the variable is logical or all are string.)

- - shared/horiz_interp/horiz_interp_bicubic.f90

  - shared/horiz_interp/horiz_interp_bicubic.html

  this module delivers methods for bicubic interpolation from acoarse regular grid on a fine regular grid.

- - shared/horiz_interp/horiz_interp.f90

  - shared/horiz_interp/horiz_interp_type.f90

  compatible with the new module horiz_interp_bicubic.f90.